

An Investigation into the Mutability of Information in Production Grid Information Systems

Laurence Field

CERN, Geneva, Switzerland

E-mail: Laurence.Field@cern.ch

Markus W. Schulz

CERN, Geneva, Switzerland

E-mail: Markus.schulz@cern.ch

Abstract. Grid Information Systems are mission-critical components in today's production Grid infrastructures. They enable users, applications and services to discover which services exist in the infrastructure and further information about their service structure and state. As the Grid Information System is pervasive throughout a Grid infrastructure, it is especially sensitive to the size of the infrastructure, and the usage of the information system increases as the infrastructure grows. Grid interoperation activities and the paradigm of multi-core processors are two main factors driving the growth of Grid infrastructures. Additional sites will bring extra resources and hence inject more information into the information system. An increasing number of cores will add to the existing execution environments available in the infrastructure, which in turn will cause an additional query load from the increased number of simultaneous computing activities that the infrastructure can support. To ensure that the current information systems are able to handle this increase in scale, it is necessary to understand where performance optimisations can be made. This paper investigates the mutability of information in the production information system of the Enabling Grid for E-Science (EGEE) project for the purpose of understanding how to best optimise the existing systems.

1. Introduction

Grid computing is defined as *coordinated resource-sharing and problem solving in dynamic, multi-institutional virtual organisations* [7]. Users who are members of Virtual Organisations can gain access to a share of a resource by using Grid services [8]. Grid Information Systems enable users, applications and services to discover which services exist in the infrastructure and further information about their structure and state. Information describing each Grid service is provided by that service itself, and an aggregated view describing many services is required to carry out specific Grid functionality.

The Enabling Grid for E-Science (EGEE) project [10] has built a production infrastructure for the European Research Area that enables coordinated resource-sharing to help solve problems in fields as diverse as high energy physics, earth and life sciences. The Grid information system used in the EGEE infrastructure is an alternative implementation of the original Meta computing Directory Service (MDS) [11] from the Globus project [6]. Within the EGEE information system,

Grid services and resources are described using the GLUE [3] information model. Information providers are used at the service-level to interrogate the service and provide information in LDIF [9] using an LDAP [12] rendering of the GLUE information model. This information is cached using an OpenLDAP [2] database at the service-level, which enables the information to be remotely queried and protects the underlying resource from external queries. This cache, along with its update mechanism, is known as the Berkeley Database Information Index (BDII) [4]. At the site-level the BDII is also used to provide a cache of the information from all the services at that site. This cache not only protects the individual services from external queries but also acts as an interface between the site and the wider Grid infrastructure. A top-level BDII is used to aggregate information from all sites participating in the EGEE project and as such gives a global overview of all the services available in the infrastructure. The top-level BDIIs provide a single point which can be queried and also protects the site-level BDIIs from end-user queries. Multiple instances of the top-level BDII are deployed to reduce the query load on each instance and to enable fault tolerance for clients that have implemented a fail-over mechanism.

The BDII update process frequently updates the database to ensure that the information is fresh. The time taken for the update defines the maximum freshness for the information in the cache. There are many factors that can affect the update time; however, previous investigations [4] have shown that the predominant factor is the amount of information. This suggests that any factors affecting the growth of the information will also affect the update time and hence freshness of the information in the database.

This paper investigates the mutability of attributes in the information system which could be used to understand if optimisations can be made that would increase the scalability limits of the current system. Section 2 outlines the method used to measure the frequency of changes and section 3 provides details of how this information was obtained along with the results. The implication of the results on the design of the information system is discussed in section 4, and the paper concludes with some concrete directions for future investigations.

2. Investigating the Mutability of Information

The growth of the EGEE production infrastructure has been measured since its initial deployment. Figure 1 shows this growth with respect to the number of sites, core and jobs.

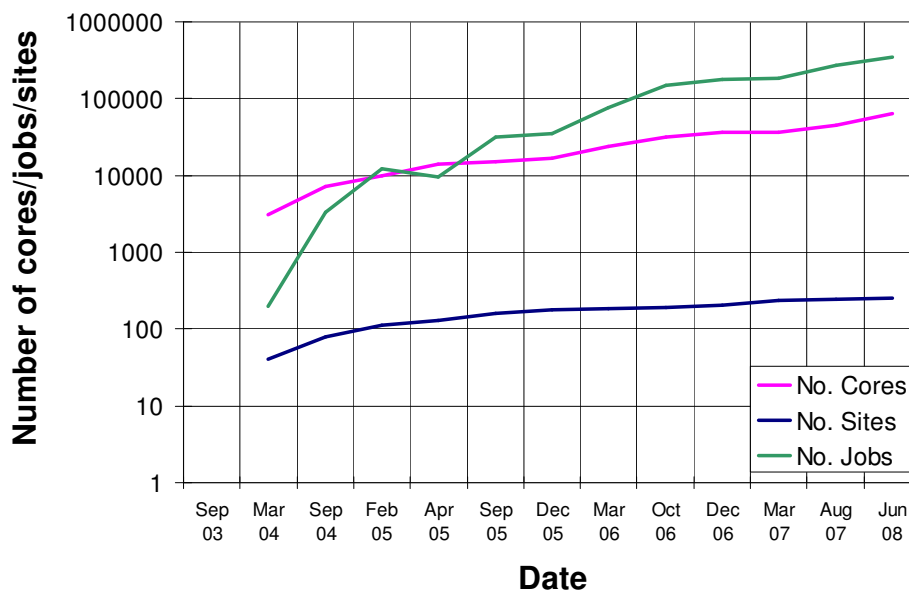


Figure 1. The Growth of the EGEE Infrastructure

It can be seen that the rate of increase with respect to the number of sites joining the infrastructure is slowing; however, the rate of increase with respect to the number of cores and jobs is increasing. As the number of research institutes in the world is fairly static, it is not surprising that the rate of growth is slowing due to the limited number of institutes available. Interoperation activities and future technological shifts, such as the introduction of campus grids [5], may increase the number of sites which are able to join the infrastructure; however this would maintain growth rather than significantly affect any existing trends. New site joining the infrastructure will bring additional Grid services and hence increase the total amount of information in the information system.

As the Grid infrastructure expands, the amount of information grows. The time taken to obtain this information and to update the database increases, which adversely affects the freshness of the information in the database. One solution to this problem would be to only update the information that has changed, a *differential update method*. The effectiveness of this approach depends on the mutability of the information. If only a few values are highly dynamic, this could lead to a significant efficiency gain. In order to measure the possible efficiency gains of using a *differential update method*, it is necessary to measure the mutability of the information.

The BDII performs two main functions; updating the information in the database and responding to queries. The update process in the BDII obtains information from a number of sources, merges the information and updates the database. The sources of information can be LDIF files, scripts that return LDIF information or other LDAP databases. Previous performance-testing of this update process [4] indicated that the most efficient method was to drop the database and repopulate it with the new information. As the update process merges the information from many sources into one LDIF file before updating the database, this file can be archived for a top-level BDII, as it represents a snapshot of the information at a point in time. Doing this for each update results in a time-series view of the information. An analysis can be carried out on this data to measure the mutability of information.

3. The Mutability of Information in Production Grid Information Systems

A top-level BDII was modified to archive the LDIF created during the update process. Over a period of nine days, the changes for 1,932 update cycles were recorded, which corresponds to approximately one update cycle every 7 minutes. Each update recorded 50MB of data which resulted in an archive of 19.6 GB, which was analysed. A graph of showing the number of modified entries per update cycle can be seen in Figure 2.

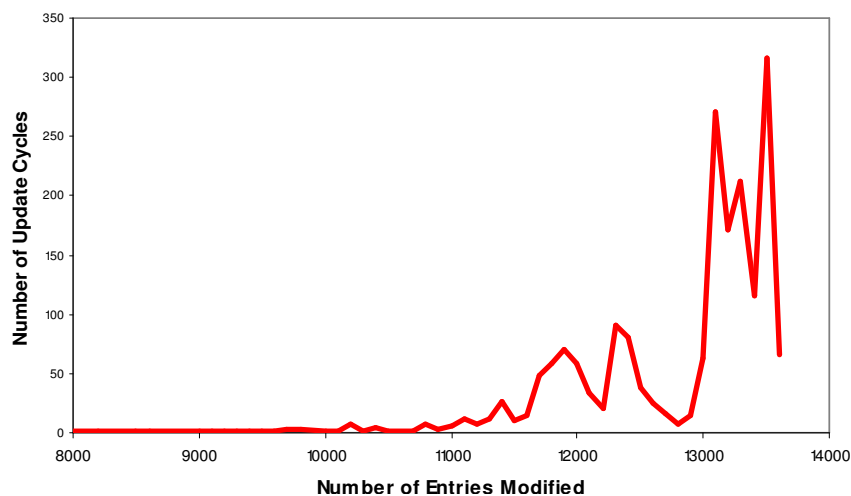


Figure 2. A graph showing the number of entries modified per update cycle.

The average number of entries modified per update cycle was 12,771, which corresponds to 21.8% of the total number of entries found in the database. This result suggests that if only the entries that contained modifications were updated, the operations required to modify the database and entries which need to be transported would be reduced by approximately 80%. For each attribute, a percentage representing its contribution to the total attributes modified over all the updates was calculated. The results for the attributes with the highest percentages are shown in Table 1. It was found that 97.8% of the changes are confined to 14 attributes which

Attribute	Percentage
GlueCEStateTotalJobs	9.41%
GlueCEStateFreeCpus	9.52%
GlueSAStateUsedSpace	5.38%
GlueCEStateFreeJobslots	19.36%
GlueCEStateWorstResponseTime	11.79%
GlueSASateAvailableSpace	6.57%
GlueCEStateEstimatedResponseTime	12.50%
GlueCEStateRunningJobs	7.90%
GlueCEInfoTotalCpus	4.67%
GlueCEStateWaitingJobs	6.37%
GlueCEPolicyAssignedJobSlots	0.90%
GlueServiceStartTime	0.71%
GlueSAUsedOnlineSize	1.34%
GlueSAFreeOnlineSize	1.37%

Table 1. An attributes contribution to the mutability as a percent

is only 4% of the total attributes used. In the current implementation, all the attributes in all entries are transported and updated during each cycle. If those 14 attributes, all of which are related to dynamic state, were removed from the information system, the changes would reduce to approximately 2% of the current amount. Combining these two factors, if the dynamic state information is removed from the existing information system and only the changes were propagated through the system, the amount of changes cascading through the system should reduce to approximately 0.5% of the current amount.

4. Discussion

The results suggests that efficiency gains could be made by optimising the system by removing highly dynamic attributes and only updating the information that has changed. As existing Grid functions rely on highly dynamic attributes in the information system, it is not possible to removed these in the short term. In the medium term it maybe possible to transport these attributes using an alternative mechanism, such as a messaging system; however, long term these attributes should be included into a monitoring system which is application-specific.

In order to achieve differential updates, it should be understood what information has changed at the source. During the update process, the BDII can compare the new LDIF file with the contents of the database to find which attributes have changed and only update the attributes that have changed. This will also result in the database having knowledge of information that has changed, which can be used in a mechanism to propagate these changes to the higher levels.

The *differential update mechanism* described above has been implemented in a new major release of the BDII, version 5.0. A top-level instance of this version was evaluated and the performance improvement was measured. Figure 3 shows the one minute load average on a

machine running BDII version 4 and the result of updating this to BDII version 5.0. It can be seen that there is a significant decrease in the load average after the machine has been upgraded, which is a result of using the *differential update mechanism*.

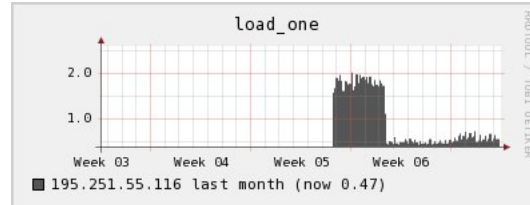


Figure 3. The one minute load average before and after upgrading to BDII version 5.0

This *differential update mechanism* carried out in the resource-level BDII distributes the task among many instances and only acts upon a small data size to the order of 50K. With the information being inserted in to the resource BDII as modifications to the database, this opens up a number of possibilities. One possibility is to use the standard OpenLDAP replication mechanism, syncrepl [1], to automatically propagate these changes to the higher levels in the system.

5. Conclusions

An aggregated view describing many services is required to carry out specific Grid functionality. Many of these use cases require this aggregated view to be up-to-date. The freshness of this view can be affected by many factors, however the amount of information is a major limiting factor. As Grid infrastructures such as EGEE continue to expand, more Grid services increase the amount of information in the information system. By understanding the mutability of the information, it was envisaged that optimisations could be made to improve the performance of the update process.

The update process used in a top-level BDII was modified to record snapshots of the information and this was used to collect data representing typical BDII updates over a nine day period. This data was analysed to investigate the mutability of the information in the information system. The results showed that on average 12,771 entries were modified per update cycle, which corresponds to 21.8% of the total number of entries found in the database. In addition, more in-depth investigation showed that 97.8% of the changes are confined to 14 attributes which is only 4% of the total attributes used.

As a result, a *differential update mechanism* was implemented in a new major release of the BDII, version 5.0, which significantly reduced the load average and update time. However, in this specific implementation, the new LDIF is generated before the difference is calculated. For the case of external sources, this results in a great deal of unnecessary information being transported over the network. Further investigation is required into efficient transport mechanisms that will only propagate the changes and not the complete information.

This investigation suggests that the majority of information changes are related to state information about services. If this information was moved to a monitoring system that specifically focused on delivering highly dynamic information, the burden on the information system which is primarily used for service discovery would be reduced.

Although this investigation has resulted in a more efficient BDII update process, there are many other areas that could be investigated which would lead to future optimisations of the existing system.

6. References

- [1] LDAP Sync Replication. <http://www.openldap.org/doc/admin22/syncrepl.html>, 2009.
- [2] The OpenLDAP Project. <http://www.openldap.org/>, 2009.
- [3] S. Androozzi, S. Burke, and L. Field and B. Konya. GLUE Schema Specification version 1.3. <http://glueschema.forge.cnaa.infn.it/Spec/V13>, 2004.
- [4] L. Field and M. W. Schulz. Grid Deployment Experiences: The path to a production quality LDAP based grid information system. In *Proceedings of the International Conference on Computing in High Energy and Nuclear Physics (CHEP 2004)*, 2004.
- [5] Ian Foster. Service-Oriented Science. *Science*, 308:814–817, 2005.
- [6] Ian Foster and Carl Kesselman. Globus: A Metacomputing Infrastructure Toolkit. *International Journal of Supercomputing Applications*, 11(2):115–128, 1997.
- [7] Ian Foster, Carl Kesselman, and Steven Tuecke. The Anatomy of the Grid: Enabling Scalable Virtual Organizations. *International Journal of High Performance Computing Applications*, 15:200–222, 2001.
- [8] Ian Foster, Carl Kesselman, and Steven Tuecke. Grid Services for Distributed System Integration. *Computer*, 35(6):37–46, 2002.
- [9] G. Good. The LDAP Data Interchange Format (LDIF) - Technical Specification. RFC 2849 (Proposed Standard), June 2000.
- [10] E. Laure and R. Jones. *Grid Computing: Infrastructure, Service, and Application*, chapter Enabling Grids for e-Science: The EGEE Project. CRC Press, 2009.
- [11] Gregor von Laszewski, Steve Fitzgerald, Ian Foster, Carl Kesselman, Warren Smith, and Steve Tuecke. A Directory Service for Configuring High-Performance Distributed Computations. *Proceedings of the 6th IEEE Symposium on High-Performance Distributed Computing*, pages 365–375, 1997.
- [12] K. Zeilenga. Lightweight Directory Access Protocol version 2 (LDAPv2) to Historic Status. RFC 3494 (Informational), March 2003.